Analysis of 2D Torus and Hub Topologies of 100Mb/s Ethernet for the Whitney Commodity Computing Testbed[1]

Kevin T. Pedretti and Samuel A. Fineberg

NAS Technical Report NAS-97-017
September 1997

MRJ, Inc.
Numerical Aerodynamic Simulation
NASA Ames Research Center, M/S 258-6
Moffett Field, CA 94035-1000

Abstract

A variety of different network technologies and topologies are currently being evaluated as part of the Whitney Project. This paper reports on the implementation and performance of a Fast Ethernet network configured in a 4x4 2D torus topology in a testbed cluster of "commodity" Pentium Pro PCs. Several benchmarks were used for performance evaluation: an MPI point to point message passing benchmark, an MPI collective communication benchmark, and the NAS Parallel Benchmarks version 2.2 (NPB2). Our results show that for point to point communication on an unloaded network, the hub and 1 hop routes on the torus have about the same bandwidth and latency. However, the bandwidth decreases and the latency increases on the torus for each additional route hop. Collective communication benchmarks show that the torus provides roughly four times more aggregate bandwidth and eight times faster MPI barrier synchronizations than a hub based network for 16 processor systems. Finally, the NPB2 benchmarks, which simulate real-world CFD applications, generally demonstrated substantially better performance on the torus than on the hub. In the few cases the hub was faster, the difference was negligible. In total, our experimental results lead to the conclusion that for Fast Ethernet networks, the torus topology has better performance and scales better than a hub based network.

1. Work performed under NASA Contract NAS 2-14303

1.0 Introduction

Recent advances in “commodity” computer technology have brought the performance of personal computers close to that of workstations. In addition, advances in “off-the-shelf” networking technology have made it possible to design a parallel system made purely of commodity components, at a fraction of the cost of MPP or workstation components. The Whitney project, being performed at NASA Ames Research Center, attempts to integrate these components in order to provide a cost effective parallel testbed.

One of the key components of Whitney is the means of interconnecting the processors. There are many custom, semi-custom, and commodity technologies available for networking. These include Ethernet, Fast Ethernet, Gigabit Ethernet, Myrinet, HiPPI, FDDI, SCI, etc. The most attractive of these choices, however, is currently Fast Ethernet, due to its good performance and extremely low cost.

Combining a large number of systems into a high performance parallel computer requires the careful selection of both network technology and topology. The Whitney project is currently evaluating different network technologies and topologies in a testbed cluster of “commodity” Intel Pentium Pro PCs. This paper will report on the implementation and performance of Fast Ethernet, both in a single hub and in a 4x4 routed 2D torus[2] topology.

The remainder of this paper is organized as follows. Section 2 will provide the configuration details for the networks we tested. In section 3, the actual hardware configuration of the testbed system will be discussed. Section 4 presents the results of the experiments. Finally, section 5 presents final conclusions along with directions for further research.

2.0 Network Configuration

Fast Ethernet [Iee95] is a ten times faster version of the original Ethernet standard. The increase of the bit rate to 100 million bits per second (Mbps) and modifications to the physical layer of the Ethernet standard are the only major changes. This has greatly helped manufacturers in bringing products to market quickly and also has created a large consumer market because of Ethernet's familiarity. As a result, the price of Fast Ethernet equipment has fallen dramatically since its introduction. A typical PCI Fast Ethernet adapter costs $50-$80, and hubs cost approximately $75 per port. In addition, because the most common physical layer for Fast Ethernet (i.e., 100baseTX) utilizes inexpensive cabling technology, category 5 unshielded twisted pair (UTP), wiring costs are also very low.

2. For the purposes of this paper, the 4x4 routed 2D torus tested will often simply be referred to as a torus.

2.1 Connection Options

To build a Fast Ethernet network, machines must be attached using either a hub, switch, or “crossover” cable. In a hub, all systems share a single broadcast network, so only one host can send and one or more hosts may receive at one time. When more than one host attempts to use the network at the same time, a “collision” occurs. The systems then retry their messages using a “carrier sense media access with collision detection” (CSMA/CD) algorithm with exponential backoff. This mechanism for handling shared network access is common to all Ethernet based systems. This means that in a hub connected system the maximum bisection bandwidth is limited to 100Mbps (12.5 MBytes/sec), and is often lower when more than one host is contending for access, regardless of the number of nodes in the network. While this is hardly adequate for a parallel system, we performed measurements on this configuration to see how it would perform.

To increase the bisection bandwidth of the system, one must increase the number of simultaneous connections possible and “break” the ethernet into multiple segments. This can be done either with an Ethernet switch or by adding TCP/IP routers. The advantage of Ethernet switching is that there still appears to be a single Ethernet network, though it will now support multiple simultaneous senders and receivers. In addition, some Ethernet switches allow nodes to operate in “Full duplex” mode where they simultaneously send and receive data. This is especially useful for acknowledgment and flow control packets that must flow from a receiver to a sender. The disadvantage, however, is that Ethernet switches are expensive, \$300-\$700 per port, and they do not scale past 100-200 nodes. Further, switches do have a limited bisection bandwidth, though they can typically deliver 1-2Gbps of aggregate bandwidth.

A second choice, however, is to utilize TCP/IP based routing where either some or all nodes forward packets between subnets. This scheme increases the aggregate bandwidth of the network without purchasing additional switching hardware (the nodes are the switches). In addition, if nodes are attached directly using “crossover” cables rather than hubs, full duplex operation is possible. However, router nodes must have more than one Ethernet card, nodes must spend CPU time forwarding packets between other nodes, and the performance of TCP/IP routing is usually lower than that of Ethernet switches.

In this paper we chose to test both a hub connected system and a routed topology. The topology we chose, a 2D torus, requires all nodes to perform routing. Further, because links are implemented with crossover cables (i.e., the network does not include any hubs), all connections can operate in full duplex mode.

The 2D torus was chosen for two reasons. The first reason was scalability, a mesh or torus network can be expanded to any size system by increasing either one or more dimension. This is particularly important because the planned size for Whitney is 400-500 nodes. In addition, by increasing both dimensions not only is the size of the mesh increased, but also the bisection bandwidth. The only limitation is that as the size increases, so does the diameter of the network. We chose to minimize this effect by keeping the mesh square and providing the wraparound connections.

The second reason for choosing a 2D torus was for physical and cost reasons. The nodes we used in the experiments had only 5 PCI slots. Utilizing single port Ethernet cards, this means that no more than 5 other systems may be attached to each node. While there are two and 4 port Ethernet cards, the per port cost is 2-4 times the cost of single port cards. Because we wanted an arbitrarily scalable network, we could not use a hypercube (we could only have up to 2^5, 32, nodes), and we would need 6 links for a 3D mesh/torus.

2.2 Torus Network

Figure 1 illustrates the 2D torus configuration. Each of the sixteen nodes was directly linked to its four nearest-neighbors via a 100 Mbs bidirectional Fast Ethernet connection. Thus, the torus was partitioned into thirty-two distinct TCP/IP subnetworks.

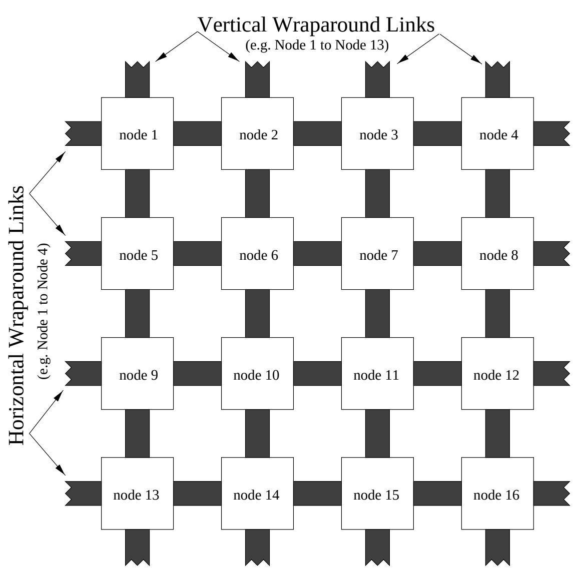

FIGURE 1. ***16 node 2D torus network configuration***

Links between neighbor interfaces (for the torus configuration) used standard category 5 unshielded twisted pair wiring that was crossed over (null modem). The wiring was tested and certified for 100 Mbs operation to ensure good connections. All links were direct so no dedicated hubs, routers, repeaters, switches, or other devices were used in the torus.

In addition to the topology depicted in Figure 1, an additional node was connected to a fifth network interface in node 1. It's major functions were to serve as a front-end for starting jobs on the cluster and to work as an NFS server for the processing nodes. Shared disk I/O, while important in a production system, was not a significant factor in any of the benchmarks which were used in this paper. The final Whitney system will have a parallel file system implemented across multiple I/O nodes.

2.2.1 Hub

For the hub experiments, all nodes were attached to two “stacked” Bay Networks Netgear FE 516 Hubs. By stacking the two 16 port hubs they act like a single 32-port hub. Each node had only a single Ethernet card and all nodes plus the front end were on a single TCP/IP subnet.

3.0 The Whitney Prototype

3.1 Hardware

The Whitney prototype consisted of 30 nodes (though only 16 were used in these experiments) with the following hardware:

- Intel Pentium Pro 200MHz/256K cache
- ASUS P/I -P65UP5 motherboard, Natoma Chipset
- ASUS P6ND CPU board
- 128 MB 60ns DRAM memory
- 2.5 GB Western Digital AC2250 hard drive
- 4 Cogent/Adaptec ANA-6911/TX ethernet cards[3]
- Trident ISA graphics card (used for diagnostic purposes only)

For this paper, we chose to concentrate on a TCP/IP routed network of systems. In addition we also performed experiments where all nodes were attached to a single Hub. Subsequent research will evaluate the cost/performance trade-offs of Ethernet switching as well as hybrid network schemes.

3.2 Software

Red Hat Linux 4.1[4] (RHL) was installed on each of the processing nodes. The kernel included with RHL, version 2.0.27, was replaced with the newest version at the time - 2.0.30. The kernel was compiled with ip forwarding turned on so that the routing mechanism of Linux could be used. Both the de4x5 v0.5 and the tulip v0.76 Ethernet drivers were tested. The de4x5 driver was used initially and exhibited some inconsistent performance characteristics. The final torus configuration on which all benchmarks were run used the tulip development driver.

A script executed at boot-time configured the Ethernet interfaces in each node. Another program set up the routing tables on each node with static routes to non-

3. Node 1 contained an additional ethernet card. The additional card was connected to the front-end node.
4. Red Hat Linux is available from http://www.redhat.com.

local subnets using an X-Y routing scheme. Packets addressed to non-neighbor nodes were forwarded through the appropriate interface towards their destination. The shortest-hop distance was maintained in all cases.

The MPI message passing system [Mes94] was used for communication between processors. MPICH (version 1.1.0) [GrL96] was the specific MPI implementation utilized. It was built using the P4 device layer, so all communication was performed on top of TCP sockets. Programs were started on the mesh by the *mpirun* program [Fin95] which resided on the front-end. *mpirun* takes the name of the program and the number of processing nodes to use and then remotely spawns the appropriate processes on the mesh. All of the benchmarks mentioned in this report used MPI for communication.

4.0 Performance

The first bechmark run on the torus measured the message latency and bandwidth of point to point links. It was useful for evaluating the performance degradation of the different route distances in the torus. The second benchmark measured the performance of collective communication. Finally, the NAS Parallel Benchmarks version 2.2 were run. These are a set of benchmarks that approximate the MPI performance of a parallel architecture on “real world” tasks (i.e., CFD codes).

4.1 Point to point message passing

To measure point-to-point message passing performance, a MPI ping-pong benchmark was utilized. This benchmark simply sent a message of a fixed size from one node to another than back. The time for this operation was then divided by two to get the time to send a message one way. The message size was varied from 1 byte to 1 Mbyte, and all experiments were repeated 20 times. Figure 2 illustrates the point to point message send/receive time from node 1 to each of the other nodes in the torus configuration. As can be seen from this graph, the message passing performance delineates itself in to 4 categories. These 4 categories represent the number of hops each node is from node 1. Therefore, the lowest transmission time is from node 1 to its adjacent neighbors, 2, 4, 5, and 13. The second category is nodes that must be communicated to through node 1’s neighbors (they are 2 hops away), i.e., 3, 6, 8, 9, 14, and 16. The third category are nodes 3 hops away, i.e., 7, 10, 12, and 15, and the final category is nodes 4 hops away for which there is only one, node 11. Similar performance curves can be generated for any other node pair, with similar results based on the nodes distance.

Figure 3 depicts the message passing time for the hub configuration. Here only a single line is shown because all nodes are of equal distance. Therefore, the performance is similar to the nearest neighbors in Figure 2.

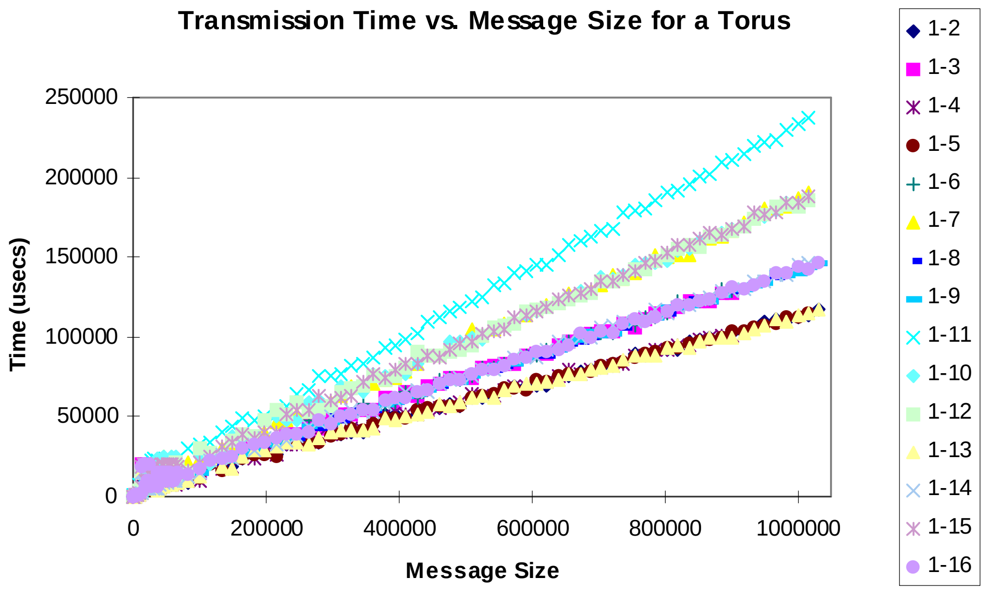

FIGURE 2. ***Point to point torus message passing time from node 1 to N***

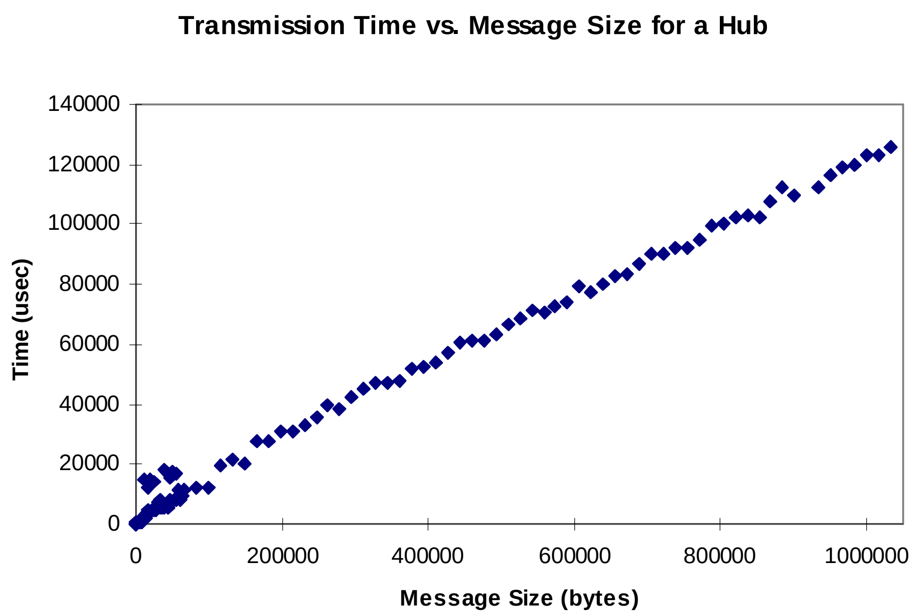

FIGURE 3. ***Point to point hub message passing performance on a hub***

4.1.1 Latency

To determine the latency of message passing Figures 4 and 5 depict the message

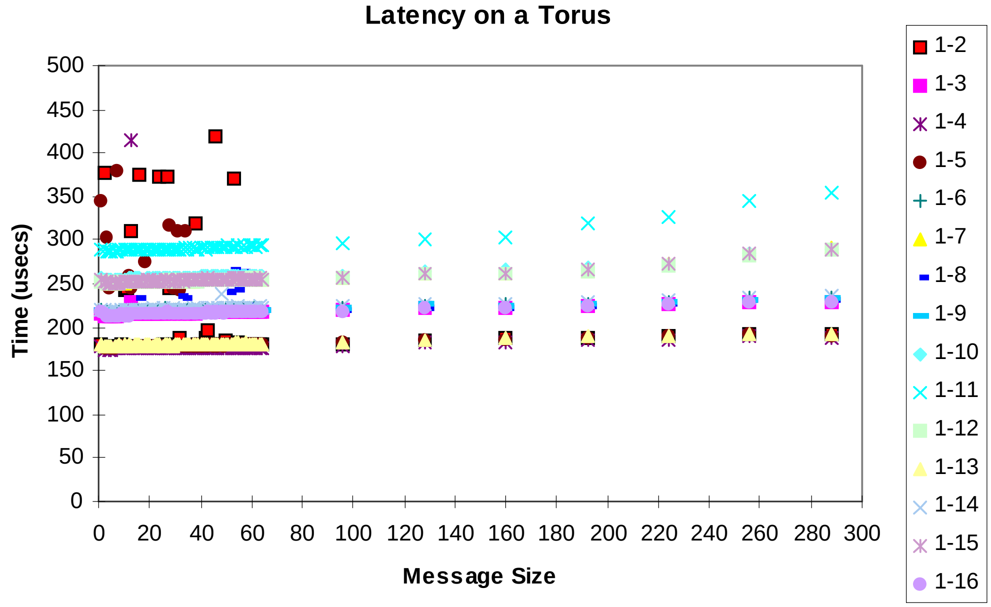

FIGURE 4. ***Transmission time for small messages on a torus***

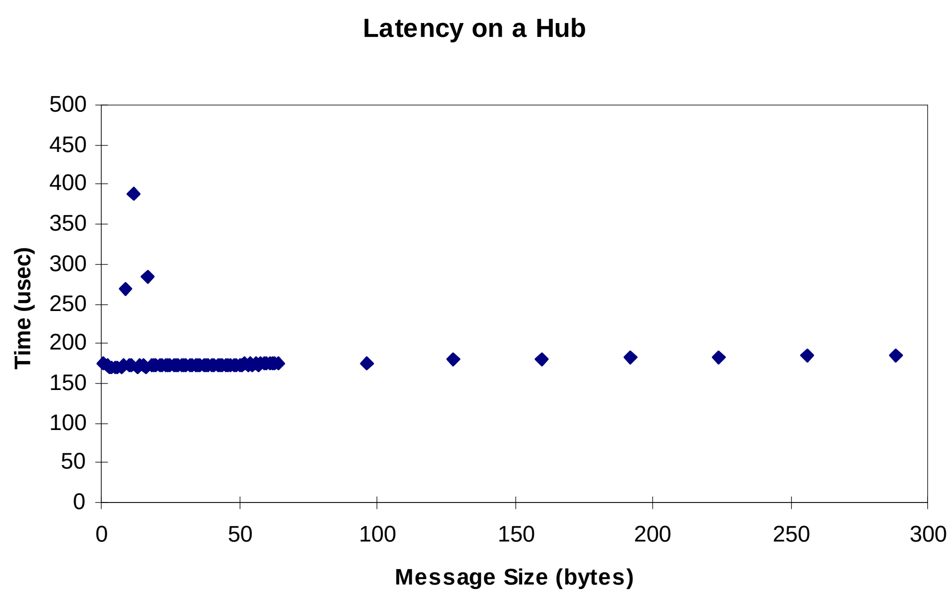

FIGURE 5. ***Transmission time for small messages on a hub***

passing time for small messages. As you can see from these graphs, latency for a single hop on the torus or for the hub are about 175 μsecs. Then, each hop on the torus adds about 40 µsecs, so the latency for 2 hops is 215 µsecs, 3 hops is 255 µsecs, and 4 hops is 295 µsecs.

4.1.2 Bandwidth

MPI Bandwidth vs. message size is shown in Figures 6 and 7. As can be seen

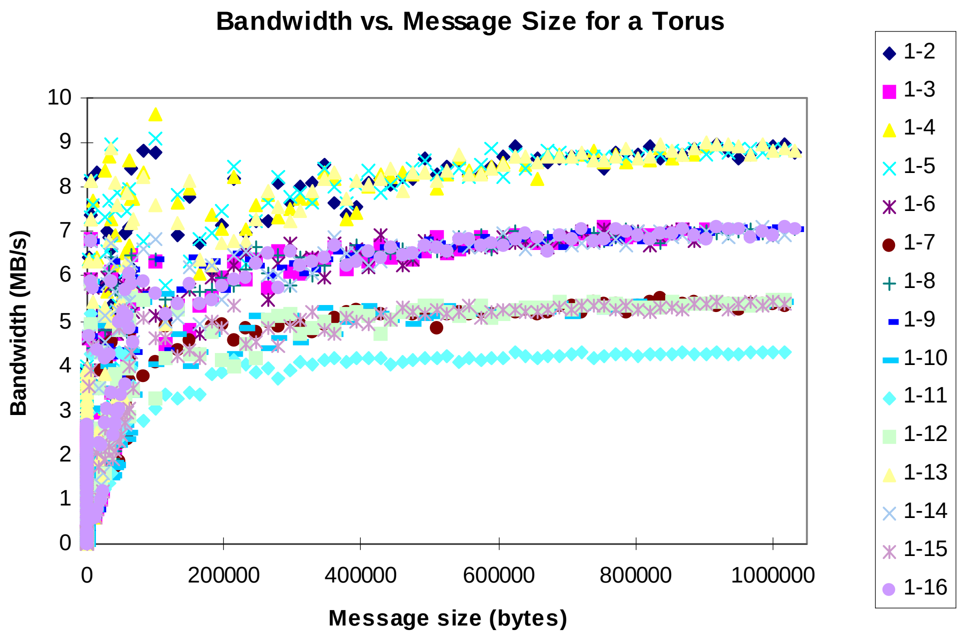

FIGURE 6. ***Bandwidth performance of the torus topology***

from these graphs, Ethernet bandwidth is quite erratic. However, some patterns can be seen. As expected, bandwidth for small message sizes is low, building to a sustained bandwidth of approximately 8-8.5 MB/sec for one hop on the torus or on the hub. For nodes more than one hop away on the torus, the bandwidth drops about 1.5 MB/sec per hop (8.5 MB/sec, 7 MB/sec, 5.5 MB/sec, 4 MB/sec). Also, note that the bandwidth reaches peak performance at an 8K message size, then it drops down and starts to build to peak slowly as message size approaches 1 MB. This anomaly is likely due to either the Ethernet or TCP packet size.

4.2 Collective Communication

To measure the performance of collective communication, a MPI broadcast benchmark was utilized. The benchmark measured the time required to broadcast a message to a given set of nodes and perform a MPI barrier synchronization. Message sizes used for the broadcast were varied between 1 and 32768 bytes in 2^n steps. Each message size was broadcast 20 times.

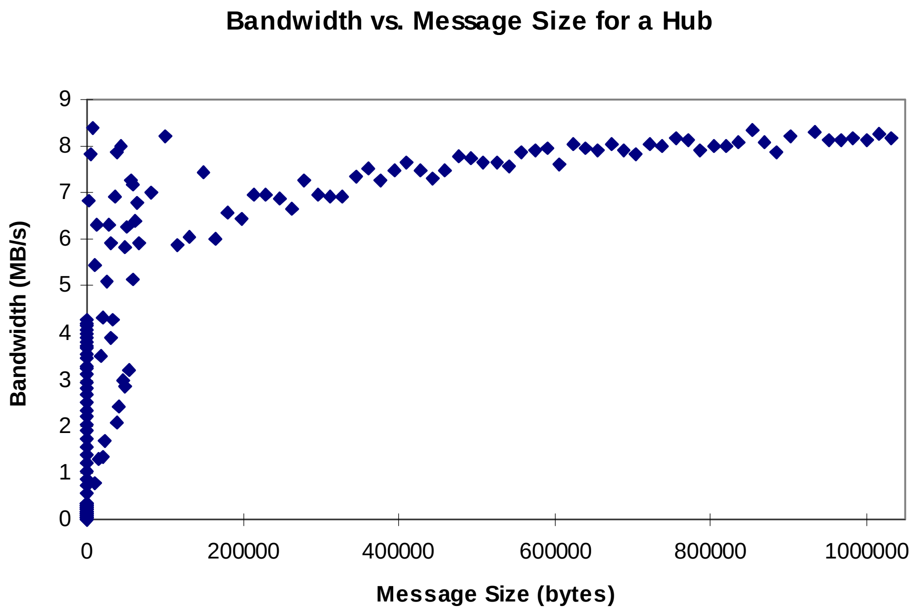

FIGURE 7. ***Bandwidth performance on a Hub***

4.2.1 Bandwidth

The aggregate bandwidth of the Torus for collective communication is depicted in Figure 8. Our experiments show that for message sizes below 1024 bytes the aggregate bandwidth is very poor. Both the Ethernet frame size and the TCP packet size could be possible causes for this. Above the 1024 byte threshold, performance becomes much closer to expected levels. The maximum aggregate bandwidth was observed to be about 43 MB/s for the 4096 byte message size. While the theoretical maximum aggregate bandwidth of the torus should be 400 MB/s, this does almost reach the maximum bisection bandwidth (50MB/s). Further, it is quite good given the cost of software routing, processor overhead, TCP/IP overhead, etc. In general, the aggregate bandwidth increases as the number of nodes increases for a given message size. The exceptions are probably due to inconsistencies in routing latency and network contention.

The collective communication bandwidth for the hub is shown in Figure 9. The cut-off for good performance is still at about 1024 bytes, however performance isn't as poor below this size as was seen in the torus. Aggregate bandwidth increases regularly as the number of nodes increases for messages up to 512 bytes. As can be seen, performance is very irregular for larger message sizes and the maximum aggregate bandwidth is about 9MB/s. Both the irregular performance and the low maximum bandwidth are probably a result of collisions on the shared 100 Mb/s network.

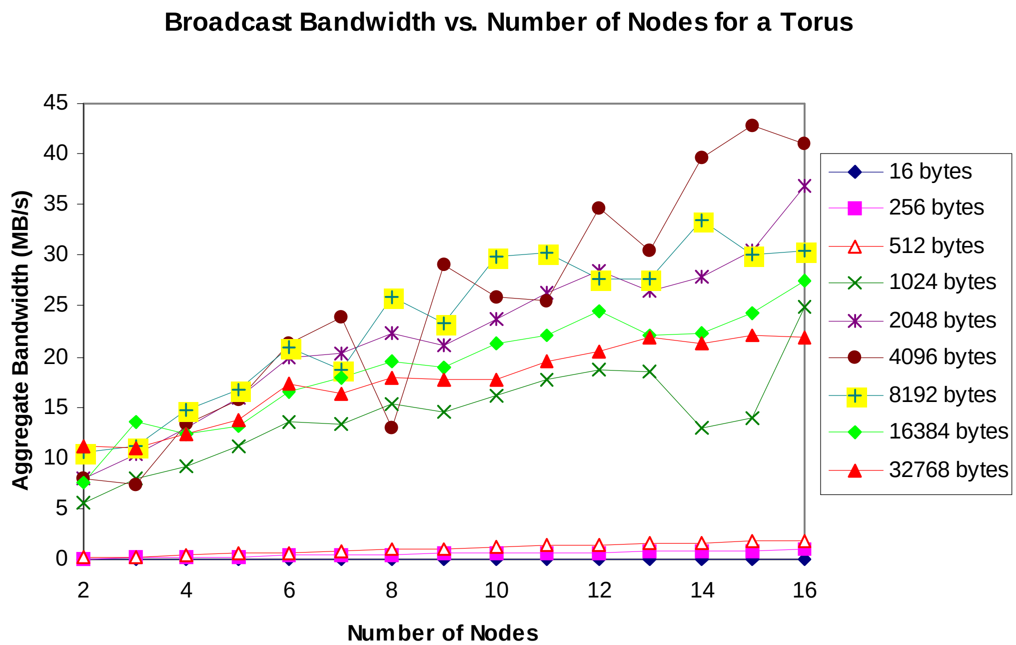

FIGURE 8. ***Collective communication performance on a torus***

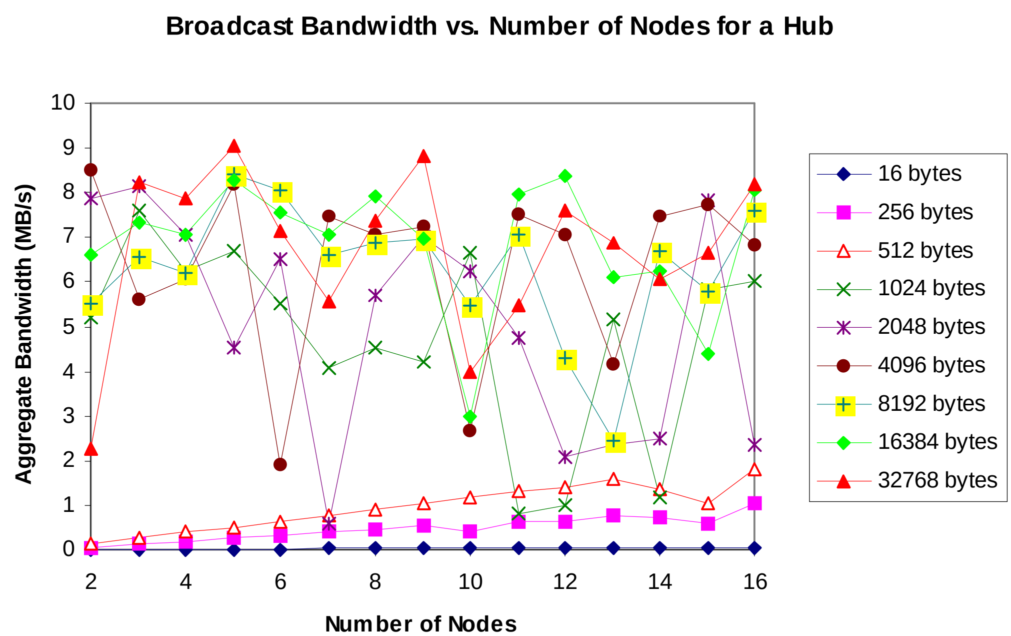

FIGURE 9. ***Collective communication performance on a hub***

4.2.2 Barrier Synchronization Time

A comparison between the hub and torus barrier synchronization time is shown in Figure 10. Clearly, the torus provides significantly faster barrier synchroniza-

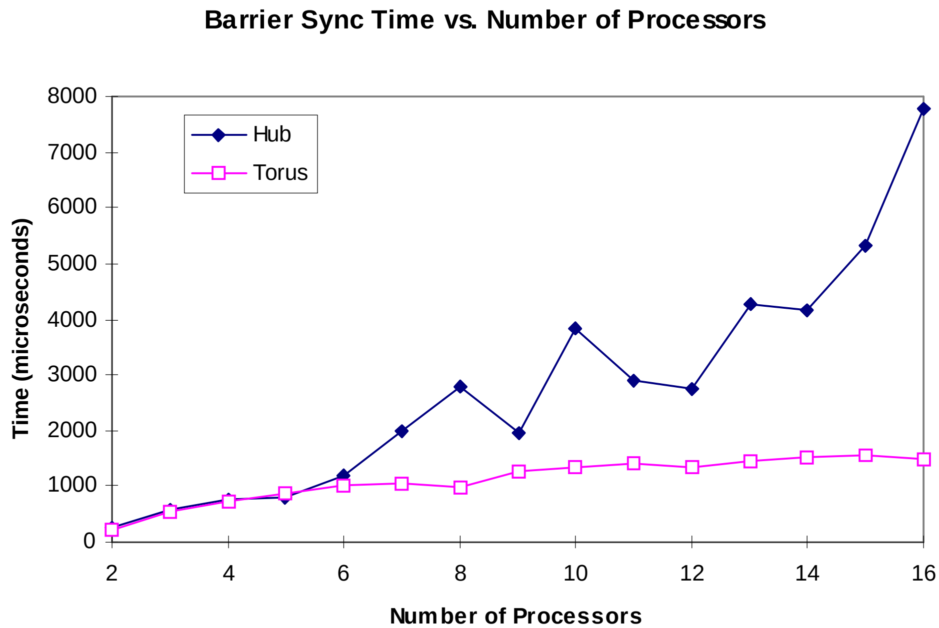

FIGURE 10. ***Comparison of barrier synchronization time for the hub and torus***

tion than the hub as the number of processors increases. Also, the hub performance is much more inconsistent. This may provide an explanation for the hubs sporadic aggregate bandwidth performance (Figure 9).

4.3 The NAS Parallel Benchmarks

Computational Fluid Dynamics (CFD) is one of the primary fields of research that has driven modern supercomputers. This technique is used for aerodynamic simulation, weather modeling, as well as other applications where it is necessary to model fluid flows. CFD applications involve the numerical solution of non-linear partial differential equations in two or three spacial dimensions. The governing differential equations representing the physical laws governing fluids in motion are referred to as the Navier-Stokes equations. The NAS Parallel Benchmarks [BaB91] consist of a set of five kernels, less complex problems intended to highlight specific areas of machine performance, and three application benchmarks. The application benchmarks are iterative partial differential equation solvers that are typical of CFD codes.

In this section, we show results for the NPB 2.2 codes [BaH95] which are MPI implementations of the NAS Parallel Benchmarks. The NPB 2.2 benchmark set includes codes for the three application benchmarks, BT, SP, and LU. It also includes code for 4 of the five original kernel benchmarks, EP, FT, MG, and IS (it

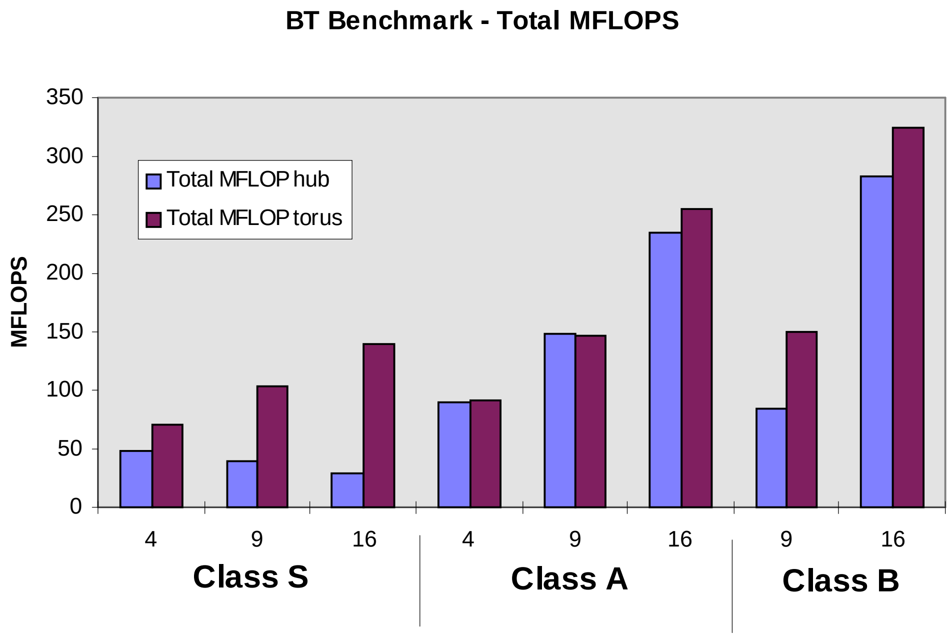

FIGURE 11. ***Comparison between hub and torus topologies for BT benchmark***

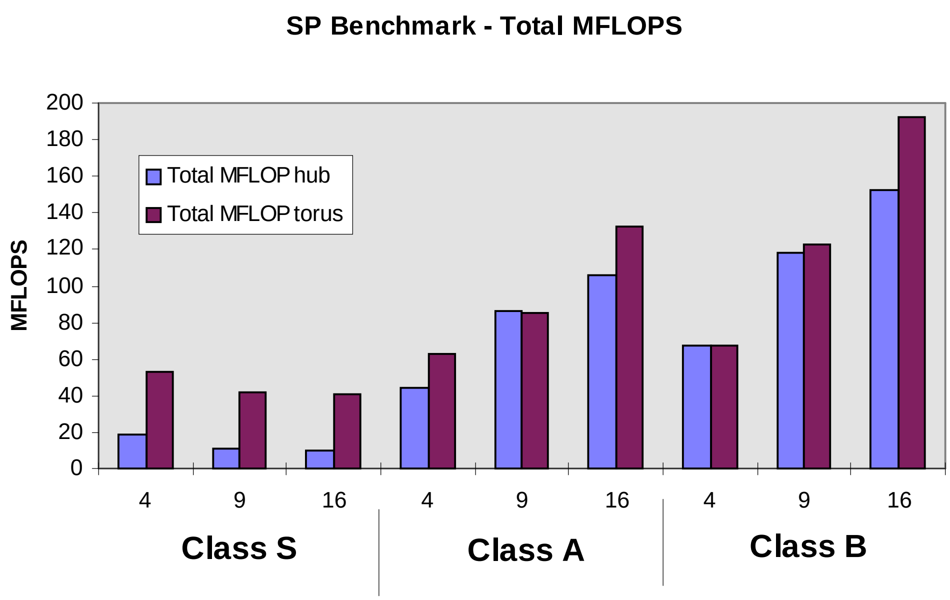

FIGURE 12. ***Comparison between hub and torus topologies for SP benchmark***

does not include CG). Full results for these codes are shown in the appendix of this paper. Benchmarks were compiled with the Portland Group's Fortran 77 compiler, pgf77, using the options: **`-O -Knoieee -Munroll -Mdalign -tp p6`**. These benchmarks were run for all valid sizes that would fit on the available nodes, This included 1, 2, 4, 8, and 16 processors for LU, FT, MG, and IS because they required processor counts that were a power of two. BT and SP,

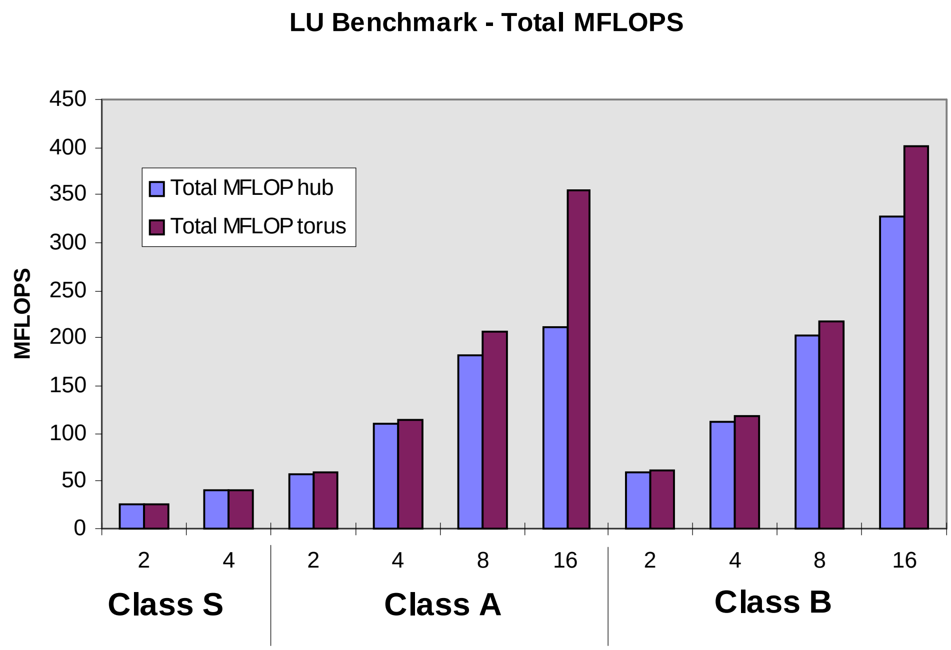

FIGURE 13. ***Comparison between hub and torus topologies for LU benchmark***

however, required sizes that were perfect squares, so they were run for 1, 4, 9, and 16 processors. Note that in the appendix single processor times are only shown for the hub, though they should be the same for the torus since the network is not used. In addition, we measured performance of the torus for both for 4 processors in a row (nodes 1, 2, 3, and 4 of Figure 1) and for a 2x2 layout (i.e., nodes 1, 2, 5, and 6 on Figure 1). This made a minor difference in performance, however it may be important on larger systems.

In Figures 11, 12, and 13 the performance of the three NAS application benchmarks on a hub and torus is compared. The first thing you will notice from the graphs is that in many cases the hub does not perform as poorly as one might expect, particularly for the Class A benchmarks. In most cases the performance of the torus was better than the hub. In the few cases where the hub was better, the difference was negligible. Also, as expected, differences between the hub and mesh increase as the number of processors increases, due to contention on the hub.

Of the application benchmarks, LU has the highest performance, 328 MFLOPS for a 16 processor hub and 402 MFLOPS for a 16 processor torus. This result is typical of the measurements we have made on Ethernet networks, i.e., LU's network characteristics seem to match nicely with Ethernet. BT also performs well, 282 MFLOPS for the hub and 323 MFLOPS for the torus, though it is significantly slower than LU. SP performs the worst, with less than half of the performance of LU. This would indicate that while some algorithms do match well to the performance characteristics of Ethernet, others perform significantly worse.

Similar differences can be seen in the kernel benchmark results as well. IS performs particularly poorly on Ethernet. EP also performs poorly in an absolute sense, but scales well, so its performance loss is likely due to the compilers or libraries available on our system. FT and MG perform reasonably, but still suffer from communication costs, especially on the hub network.

5.0 Conclusion

Our experimental results have shown that in the Whitney testbed cluster, a Fast Ethernet torus exhibits more desirable performance characteristics than a Fast Ethernet hub network. Collective communication tests show that for a 16 processor system, aggregate bandwidth is more than 4 times higher on a torus than a hub. Furthermore, many of the NPB2 results for the torus showed significant performance increases over the hub as the problem size and the number of processors were scaled. No NPB2 result showed more than a negligible performance advantage for using the hub.

There is also strong evidence that a torus will scale much more regularly than a hub network for larger processor numbers. Our results have shown that collective communication bandwidth is very sporadic on the hub relative to the torus performance. Also, the MPI barrier synchronization time for the torus was shown to scale more regularly and be much less than in a hub topology. These results are evidence of the hub's shared 100 Mb/s bandwidth becoming overloaded. The fine-grained segmentation of the torus largely prevents this problem.

In conclusion, Fast Ethernet configured in a torus topology has been shown to have better performance and to scale better than a hub based network. Future studies are planned to evaluate other network technologies and topologies in the Whitney testbed cluster. Fast Ethernet switching, for example, is interesting because it eliminates the processing load of software routing, provides a high degree of network segmentation, and has the ease of use of a hub. Myrinet is another promising network technology planned for evaluation. This paper provides a useful foundation on which to make these future comparisons.

6.0 References[5]

[BaB91] D.H. Bailey, J. Barton, T.A. Lasinski, and H. Simon, *The NAS Parallel Benchmarks*, Tech. Report RNR-91-002, NASA Ames Research Center, 1991.

5. NAS technical reports are available via the WWW at URL: **`http://www.nas.nasa.gov`**

[BaH95] D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, and M. Yarrow, *The NAS Parallel Benchmarks 2.0*, Tech. Report NAS-95-020, NASA Ames Research Center, 1995.

[Fin95] S.A. Fineberg, *Implementing Multidisciplinary and Multi-zonal Applications Using MPI*, Tech. Report NAS-95-003, NASA Ames Research Center, 1995.

[GrL96] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, *A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard*, http://www.mcs.anl.gov/mpi/mpicharticle/paper.html, Argonne National Laboratory and Mississippi State Univ., 1996.

[Iee95] *802.3u-1995 Supplement to CSMA/CD: MAC Parameters, Physical Layer, and MAUs*, Institute of Electrical and Electronic Engineers, 1995.

[Mes94] Message Passing Interface Forum, *MPI: A Message Passing Interface Standard*, Computer Science Dept. Technical Report CS-94-230, University of Tennessee, 1994.

Appendix: NAS Parallel Benchmark Results

Benchmark	Procs	Class	Hub			Torus		
			Time	MF(total)	MF/node	Time	MF (total)	MF/node
BT	1	S	7.77	29.40	29.40			
	4	S	4.78	47.77	11.94	3.24	70.36	17.59
	2x2	S	N/A	N/A	N/A	3.24	70.53	17.63
	9	S	5.68	40.22	4.47	2.20	103.89	11.54
	16	S	7.95	28.73	1.80	1.63	139.80	8.74
	1	A	7110.47	23.67	23.67			
	4	A	1873.15	88.84	22.46	1872.80	89.86	22.46
	2x2	A	N/A	N/A	N/A	1848.78	91.02	22.76
	9	A	1136.36	148.09	16.45	1152.06	146.07	16.23
	16	A	715.25	235.28	14.70	657.91	255.79	15.99
	9	B	8254.56	85.07	9.45	4684.24	149.90	16.66
	16	B	2487.79	282.25	17.64	2171.84	323.31	20.21
SP	1	S	3.16	30.63	30.63			
	4	S	5.13	18.86	4.72	1.84	52.57	13.14
	9	S	8.64	11.19	1.24	2.32	41.67	4.63
	16	S	9.68	9.99	0.62	2.34	41.35	2.58
	1	A	4480.75	18.97	18.97			
	4	A	1943.83	43.73	10.93	1360.48	62.49	15.62
	2x2	A	N/A	N/A	N/A	1347.95	63.07	15.77
	9	A	991.09	85.77	9.53	997.46	85.23	9.47
	16	A	805.51	105.54	6.60	638.64	133.11	8.32
	1	B	19119.02	18.57	18.57			
	4	B	5272.22	67.34	16.83	5282.08	67.21	16.80
	9	B	2992.99	118.61	13.18	2889.41	122.87	13.65
	16	B	2330.12	152.36	9.52	1850.76	191.82	11.99
LU	2	S	4.00	25.56	12.78	3.99	25.65	12.82
	4	S	2.58	39.71	9.93	2.56	40.01	10.00
	2x2	S	N/A	N/A	N/A	2.55	40.05	10.01
	2	A	2058.19	57.96	28.98	2003.39	59.55	29.77
	4	A	1081.56	110.30	27.58	1058.95	112.66	28.16
	2x2	A	N/A	N/A	N/A	1038.78	114.84	28.71
	8	A	656.14	181.82	22.73	577.39	206.61	25.83
	16	A	566.64	210.53	13.16	335.69	355.37	22.21
	2	B	8387.14	59.47	29.74	8249.52	60.47	30.23
	4	B				4266.09	116.93	29.23
	2x2	B	N/A	N/A	N/A	4222.93	118.12	29.53
	8	B	2472.05	201.79	25.22	2285.74	218.23	27.28
	16	B	1520.33	328.10	20.51	1240.17	402.22	25.14

Benchmark	Procs	Class	Hub			Torus		
			Time	MF(total)	MF/node	Time	MF (total)	MF/node
	1	S	59.94	0.56	0.56			
	2	S				30.00	1.12	0.56
	4	S	15.05	2.23	0.56	15.08	2.23	0.56
	2x2	S	N/A	N/A	N/A	15.09	2.22	0.56
	8	S	7.55	4.45	0.56	7.54	4.45	0.56
	16	S	4.53	7.40	0.46	4.00	8.38	0.52
	1	A	958.95	0.56	0.56			
	2	A				479.50	1.12	0.56
EP	4	A	239.82	2.24	0.56	239.83	2.24	0.56
	2x2	A	N/A	N/A	N/A	239.82	2.24	0.56
	8	A	120.03	4.47	0.56	119.90	4.48	0.56
	16	A	60.02	8.94	0.56	60.73	8.84	0.55
	1	B	3835.95	0.56	0.56			
	2	B				1918.30	1.12	0.56
	4	B	959.19	2.24	0.56	958.81	2.24	0.56
	2x2	B	N/A	N/A	N/A	960.26	2.24	0.56
	8	B	479.73	4.48	0.56	480.58	4.47	0.56
	16	B	239.82	8.95	0.56	240.43	8.93	0.56
	1	S	0.32	23.48	23.48			
	2	S	0.34	22.68	11.34			
	4	S	1.19	6.40	1.60	0.34	22.10	5.52
	2x2	S	N/A	N/A	N/A	0.33	22.81	5.70
	8	S	1.54	4.93	0.62	0.19	40.02	5.00
	16	S	1.39	5.48	0.34	0.15	51.55	3.22
	1	A	176.73	22.02	22.02			
MG	4	A	+	+	+	54.46	71.48	17.87
	2x2	A	N/A	N/A	N/A	52.56	74.06	18.51
	8	A	35.63	109.25	13.66	9.44	132.23	16.53
	16	A	26.41	147.37	9.21	17.52	222.21	13.89
	1	B	829.01	23.48	23.48			
	4	B	+	+	+	257.67	75.53	18.88
	2x2	B	N/A	N/A	N/A	248.30	78.38	19.60
	8	B	156.78	124.13	15.52	128.00	152.05	19.01
	16	B	125.34	155.27	9.70	82.76	235.16	14.70

Benchmark	Procs	Class	Hub Time	Hub MF(total)	Hub MF/node	Torus Time	Torus MF (total)	Torus MF/node
IS	1	S	0.20	3.27	3.27			
	2	S	0.30	2.17	1.09	0.34	1.92	0.96
	4	S	0.37	1.76	0.44	0.45	1.44	0.36
	2x2	S	N/A	N/A	N/A	0.37	1.75	0.44
	8	S	0.67	0.97	0.12	0.12	5.56	0.69
	16	S	0.70	0.94	0.06	0.18	3.68	0.23
	1	A	30.94	2.71	2.71			
	2	A	32.37	2.59	1.30	30.85	2.72	1.36
	4	A	34.13	2.46	0.61	32.00	2.62	0.66
	2x2	A	N/A	N/A	N/A	31.88	2.63	0.66
	8	A	33.95	2.47	0.31	26.46	3.17	0.40
	16	A	40.26	2.08	0.13	34.96	2.40	0.15
	1	B	129.02	2.60	2.60			
	4	B	138.73	2.42	0.60	133.27	2.52	0.63
	2x2	B	N/A	N/A	N/A	122.26	2.74	0.69
	8	B	132.71	2.53	0.32	94.03	3.57	0.45
	16	B	133.91	2.51	0.16	83.12	4.04	0.25
FT	1	S	6.65	26.66	26.66			
	2	S				5.64	31.39	15.69
	4	S	4.49	39.46	9.86	4.49	39.46	9.86
	2x2	S	N/A	N/A	N/A	4.26	41.54	10.39
	8	S	4.46	39.75	9.94	5.87	30.19	3.77
	16	S	6.60	26.86	1.68	6.30	28.10	1.76
	1	A	272.32	26.21	26.21			
	2x2	A	N/A	N/A	N/A	270.98	26.34	6.58
	4	A	261.86	27.25	6.81	251.00	28.43	7.11
	8	A	113.37	62.95	7.87	95.57	74.67	9.33
	16	A	99.80	71.51	4.47	64.31	110.98	6.94
	16	B	2122.13	43.38	2.71	1922.18	47.89	2.99

<table>
<tr><td rowspan="2"></td><td>NAS TECHNICAL REPORT</td></tr>
<tr><td>Title:
Analysis of 2D Torus and Hub Topologies of 100Mb/s Ethernet for the Whitney Commodity Computing Testbed</td></tr>
<tr><td></td><td>Author(s):
Kevin T. Pedretti and Samuel A. Fineberg</td></tr>
<tr><td>Two reviewers must sign.</td><td>Reviewers:
“I have carefully and thoroughly reviewed this technical report. I have worked with the author(s) to ensure clarity of presentation and technical accuracy. I take personal responsibility for the quality of this document.”

Signed: ______________________________

Name: _Jeff Becker_________________

Signed: ______________________________

Name: _Dave Tweten_______________</td></tr>
<tr><td>After approval, assign NAS Report number.</td><td>Branch Chief:

Approved: ______________________________</td></tr>
<tr><td>Date:</td><td>NAS Report Number:</td></tr>
</table>